



Joonas Partanen

## **MAEMO 5 JA WINDOWS PHONE 7 – SOVELLUSKEHITYSPROSESSIEN VERTAILU**

## **MAEMO 5 JA WINDOWS PHONE 7 – SOVELLUSKEHITYSPROSESSIEN VERTAILU**

Joonas Partanen  
Opinnäytetyö  
26.8.2011  
Tietotekniikan koulutusohjelma  
Oulun seudun ammattikorkeakoulu

# TIIVISTELMÄ

## OULUN SEUDUN AMMATTIKORKEAKOULU

## TIIVISTELMÄ

Koulutusohjelma	Opinnäytetyö	Sivuja	+	Liitteitä
Tietotekniikka	Opinnäytetyö	41	+	0
Suuntautumisvaihtoehto	Aika			
Ohjelmistojen kehitys	2011			
Työn tilaaja	Työn tekijä			
Nokia Oyj, Devices R&D	Joonas Partanen			
Työn nimi				
Maemo 5 ja Windows Phone 7 – Sovelluskehitysprosessien vertailu				
Avainsanat				
Maemo, Qt, QML, WP7, Windows Phone 7, XAML, C#, ohjelmistokehitys				

Opinnäytetyön tarkoituksena oli vertailla Maemo- ja Windows Phone 7 - mobiilikäyttöjärjestelmiä sovelluskehittäjän näkökulmasta. Työn tavoitteena oli toteuttaa musiikkisoitinsovellus molemmille alustoille ja samalla tutustua alustoihin ja niiden tarjoamiin mahdollisuuksiin.

Työn ensimmäisessä vaiheessa toteutettiin musiikkisoitinsovellus Maemolle, koska sille sovelluskehitys oli jo ennestään tuttua, ja toisessa vaiheessa vastaavilla toiminnallisuuksilla varustettu soitin Windows Phone 7:lle. Työn raportointi on jaettu kolmeen osaan. Ensimmäisessä osassa on esitelty työn toteutuksessa käytettävät tekniikat, toisessa osassa on esitetty itse työn toteutus ja kolmannessa osassa vertaillaan alustoja työn aikana kertyneiden kokemusten perusteella.

Työn tuloksena saatiin toteutettua lähes samoilla toiminnallisuuksilla varustetut musiikkisoittimet molemmille alustoille. Työn toteutuksen aikana myös muodostui selkeä näkemys molemmista alustoista, niiden eroista ja niiden tarjoamista mahdollisuuksista. Molemmista alustoista paljastui sekä hyviä että huonoja puolia, ja onkin mahdollisuus suoranaisesti sanoa kumpaan alustaa paremmaksi.

# ABSTRACT

## OULU UNIVERSITY OF APPLIED SCIENCES

## ABSTRACT

Degree programme	Thesis	Pages	+	Appendices
Information Technology	Bachelor thesis	41	+	0
Line	Date			
Software development	2011			
Comissioned by	Author			
Nokia Oyj, Devices R&D	Joonas Partanen			
Thesis title				
Comparing the software development process between Maemo 5 and Windows Phone 7				
Keywords				
Maemo, Qt, QML, WP7, Windows Phone 7, XAML, C#, software development				

The purpose of this bachelor's thesis was comparing the software development process between Maemo and Windows Phone 7 from application developer's point of view. The objective was to implement a music player for both platforms.

Work in the first phase was the music player for Maemo and on the next phase for Windows Phone 7. This document is divided into three parts. At the first part is presented all the techniques used on the project. At the second part is presented the development process of both music players and the third part contains a comparison of platforms.

Result of the work became a music player for both platforms and both have basically same functionality. During the execution of the work also consisted of a clear vision of platforms, their differences and the opportunities they offer. Both platforms revealed the strengths and weaknesses and it's hard to say which one is better.

# ALKULAUSE

Tämä opinnäytetyö on Nokia Oyj:lle tehty työ, jossa vertaillaan sovelluskehitysprosessia Maemo- ja Windows Phone 7 -käyttöjärjestelmälle. Työn ohjaajana Nokialla toimi product manager Aarne Taube, ohjaavana opettajana lehtori Eero Nousiainen ja kielenohjaajana lehtori Tuula Hopeavuori. Kiitokset työn aiheen tarjoamisesta kuuluvat Nokialle ja koko työn valmistumisesta kaikille yllä mainituille henkilöille.

Oulussa 26.8.2011

Joonas Partanen

# SISÄLLYS

TIIVISTELMÄ .....	3
ABSTRACT .....	4
ALKULAUSE .....	5
SISÄLLYS .....	6
SANASTO .....	8
1 JOHDANTO.....	9
2 TAUSTAT JA LÄHTÖKOHDAT .....	10
3 OHJELMISTOKEHITYS MAEMOLLE .....	11
3.1 Maemo.....	11
3.2 Menetelmät ja työkalut .....	12
3.3 Julkaisu.....	14
4 OHJELMISTOKEHITYS WINDOWS PHONE 7:LLE .....	17
4.1 Windows Phone 7 -käyttöjärjestelmä.....	17
4.2 Menetelmät ja työkalut.....	18
4.3 Julkaisu.....	20
5 SCRUM – KETTERÄ KEHITYSMENETELMÄ .....	21
5.1 Scrum-prosessi .....	21
5.2 Roolit.....	22
5.3 Dokumentit.....	23
6 TOTEUTUS .....	27
6.1 Vaatimusmäärittely .....	27
6.2 Suunnittelu .....	28
6.3 Maemo 5.....	29
6.3.1 Arkkitehtuuri .....	29
6.3.2 Toteutus .....	30
6.3.3 Testaus .....	32
6.3.4 Julkaisu .....	33
6.4 Windows Phone 7 .....	33
6.4.1 Arkkitehtuuri .....	33
6.4.2 Toteutus .....	34

6.4.3 Testaus .....	35
6.4.4 Julkaisu .....	36
7 VERTAILU .....	37
7.1 Kehitystyökalut.....	37
7.2 Toteutus.....	37
7.3 Julkaisu.....	38
8 YHTEENVETO .....	39
LÄHTEET .....	40

## SANASTO

**DreamSpark** Palvelu joka tarjoaa Microsoftin kehitystyökalut ilmaiseksi opiskelijoille

**GTK+** GIMP Toolkit. Graafinen käyttöliittymäkirjasto, jolla voidaan toteuttaa sovelluksiin käyttöliittymiä useille eri käyttöjärjestelmille

**.NET** Microsoftin kehittämä ohjelmistokomponenttikirjasto

**QML** Kuvauskieli käyttöliittymien toteuttamiseen Qt-sovelluksille

**Qt** Alustariippumaton sovelluskehitysympäristö

**SDK** Software Development Kit. Paketti joka sisältää kaikki sovelluskehityksessä tarvittavat työkalut

**WP7** Windows Phone 7 -mobiilikäyttöjärjestelmä

**XAML** Extensible Application Markup Language. Microsoftin kehittämä struktuurinen kuvauskieli



# 1 JOHDANTO

Työn tarkoituksena oli vertailla sovelluskehitysprosessia Maemo- ja Microsoft Windows Phone 7 - mobiilikäyttöjärjestelmien välillä Maemo-osaajan näkökulmasta. Työn ensimmäisessä vaiheessa toteutettiin musiikkisoitin-sovellus Maemo 5 -käyttöjärjestelmälle ja toisessa vaiheessa samoilla toiminnallisuuksilla varustettu musiikkisoitinsovellus Microsoft Windows Phone 7 -käyttöjärjestelmälle (WP7). Sovelluskehityksestä Maemolle oli jo kertynyt runsaasti kokemusta aikaisemmin harjoitteluprojektien kautta, mutta Windows Phone 7 -ohjelmointi oli käytännössä kokonaan uusi ympäristö ja vaati opiskelua. Sovelluskehitysprosessin vertailussa vertailtiin ohjeita, dokumentaatiota, työkaluja ja kaikkea, mikä vaikuttaa sovelluskehityksen helppouteen ja houkuttelevuuteen.

Tarkoituksena oli toteuttaa musiikkisoitinsovellus Maemolle Qt:n ja QML:n avulla. Toiminnallisuudet toteutettiin käyttämällä ohjelmointikielenä C++:aa ja käyttöliittymän toteutukseen käytettiin QML:ää. Windows Phone 7:lle sovellus oli tarkoitus toteuttaa Silverlightin avulla ja ohjelmointikielenä sille oli C#.

Tämä dokumentti on jaettu kolmeen osaan. Ensimmäisessä osassa on esitelty työn lähtökohtia, molemmat alustat ja ketterä kehitysmenetelmä Scrum. Toisessa osassa on kuvattu musiikkisoitinsovellusten toteutusvaihe molemmille alustoille ja sen jälkeen kolmannessa osassa olen vertaillut alustoja omien mielipiteiden ja kokemusten pohjalta.

## 2 TAUSTAT JA LÄHTÖKOHDAT

Työ oli jatkoa opiskeluihin liittyville harjoitteluprojekteille, jotka olin tehnyt samassa paikassa. Aiemmissa projekteissa kohdeympäristönä oli Maemo 5, jolle toteutettiin sovelluksia Qt:n avulla ensimmäiset kaksi projektia, ja kolmannessa projektissa käyttöön otettiin myös QML-kuvauskieli käyttöliittymien toteutukseen.

Idea työn aiheelle syntyi, kun Nokia ilmoitti strategiamuutoksestaan ja otti Windows Phone 7 -käyttöjärjestelmän (WP7) ensisijaiseksi strategiaksi älypuhelimissaan. Sen vuoksi katsottiin, että oli ajankohtainen aihe vertailla Maemo 5:n ja Windows Phone 7:n sovelluskehitysprosesseja, tarjolla olevia työkaluja ja kaikkea, mikä liittyy sovelluskehitykseen kyseessä oleville alustoille.

WP7:lle ohjelmoinnista ei ollut kokemusta aikaisemmin kertynyt, mutta vanhemmalle Windows Mobile 6 -käyttöjärjestelmälle oli koulussa joitain harjoitustöitä tehty. Koulussa tehdyissä harjoitustöissä ei kuitenkaan käytetty uudemmassa käyttöjärjestelmässä käytössä olevaa XAML-kuvauskieltä käyttöliittymien toteutukseen, joten sen joutui työn aikana opettelemaan perusteista alkaen. Lähtökohtina siis Qt-ohjelmointi (ja QML) oli hyvin hallussa ja C#-ohjelmoinnista olivat perusteet hallussa, mutta vähän päässeet unohtumaan.

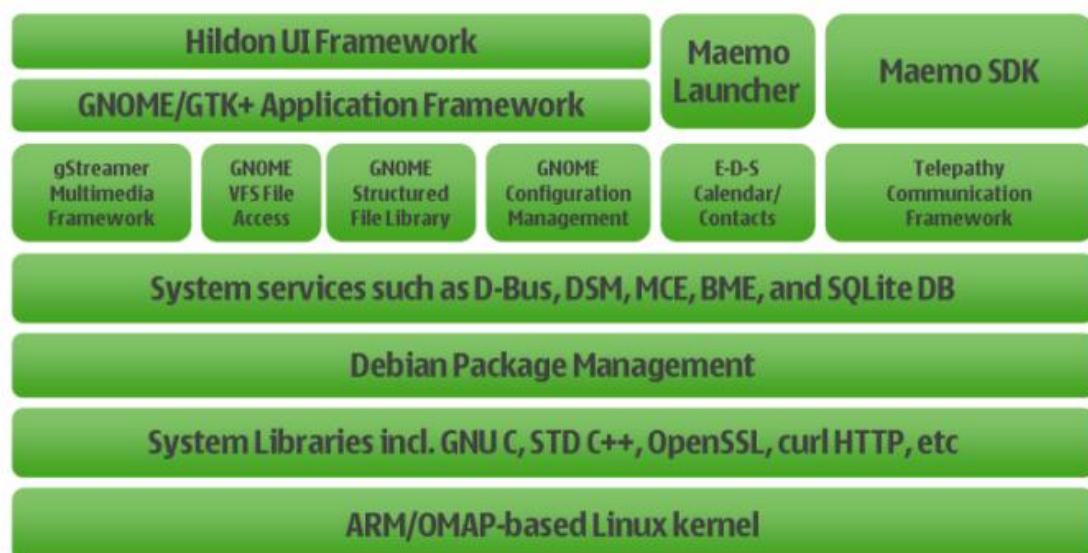
Työn toteutusvaiheessa oli tavoitteena tutustua ketterään kehitysmenetelmään, Scrumiin. Scrum-prosessia käytettiin hieman soveltaen, koska työn tekemiseen ei osallistunut suurempaa tiimiä. Esimerkiksi päivittäiset Scrum-palaverit ja tehtävien toteutukseen kuluvan ajan arvioinnit jätettiin pois kokonaan.

### 3 OHJELMISTOKEHITYS MAEMOLLE

Tässä luvussa esitellään Maemo-käyttöjärjestelmä, menetelmät ja työkalut sovellusten toteuttamiseen sekä Maemo-sovellusten julkaisuprosessi.

#### 3.1 Maemo

Maemo on Nokia Oyj:n aloittaman ja Maemo-yhteisön vetämän avoimen lähdekoodin projektin (open-source project) kehittäessä Linux-pohjainen mobiilikäyttöjärjestelmä, joka on suunniteltu toimimaan high-end-mobiilitietokoneilla, kuten Nokia N900 (Maemo background. 2011). Maemo koostuu suurimmaksi osaksi avoimen lähdekoodin periaatteella toteutetuista osista. Käyttöjärjestelmän päällä pyörii Hildon-niminen käyttöliittymä, joka on optimoitu toimimaan kosketusnäytöllä varustetuissa laitteissa. Koska kyseessä on avoimen lähdekoodin projekti, voi koko Maemo-yhteisö muuttaa ja tuottaa uusia ominaisuuksia Maemolle. Maemon eri hierarkiatasot on esitetty kuvassa 1. (Maemo.org. 2011, linkit Intro -> Software Platform.)



KUVA 1. Maemon hierarkiatasot (Maemo.org. 2011, linkit Intro -> Software Platform)

Ensimmäinen Maemo-käyttöjärjestelmällä varustettu laite julkaistiin marraskuussa 2005. Laite oli Nokia 770 Internet Tablet. Laite mahdollisti internetin käytön WLANin kautta. Seuraava Maemo-laite oli Nokia N800 Internet Tablet, joka käytti Maemo 3 -versiota. Laite mahdollisti Skype/VoIP-puhelut WLANin kautta. Seuraava Maemoa käyttävä laite oli Nokia N810, joka käytti Maemon 4. versiota. Laitteessa oli täysi QWERTY-näppäimistö, Mozilla-pohjainen internet-selain, GPS ja tuki GoogleTalkille. (Maemo.org. 2011, linkit Intro -> Evolution Of Maemo.)

Maemo 5 oli iso askel Maemon kehityksessä. Viidenteen versioon Maemon käyttöliittymä uudistettiin kokonaan toimimaan paremmin kosketusnäytöllisissä laitteissa. Maemoon lisättiin myös mahdollisuus puheluiden soittamiseen matkapuhelinverkon kautta ja lisäksi järjestelmä mahdollisti moniajon. Ensimmäinen Maemo 5 -laite oli Nokia N900, joka on ensimmäinen Maemo-laite, joka oli samalla myös puhelin. Siinä on tehokas Cortex-A8-suoritin ja OpenGL ES 2.0 -grafiikkapiiri. Laitteella on mahdollista pitää auki useita sovelluksia samaan aikaan. (Maemo.org. 2011, linkit Intro -> Evolution Of Maemo.)

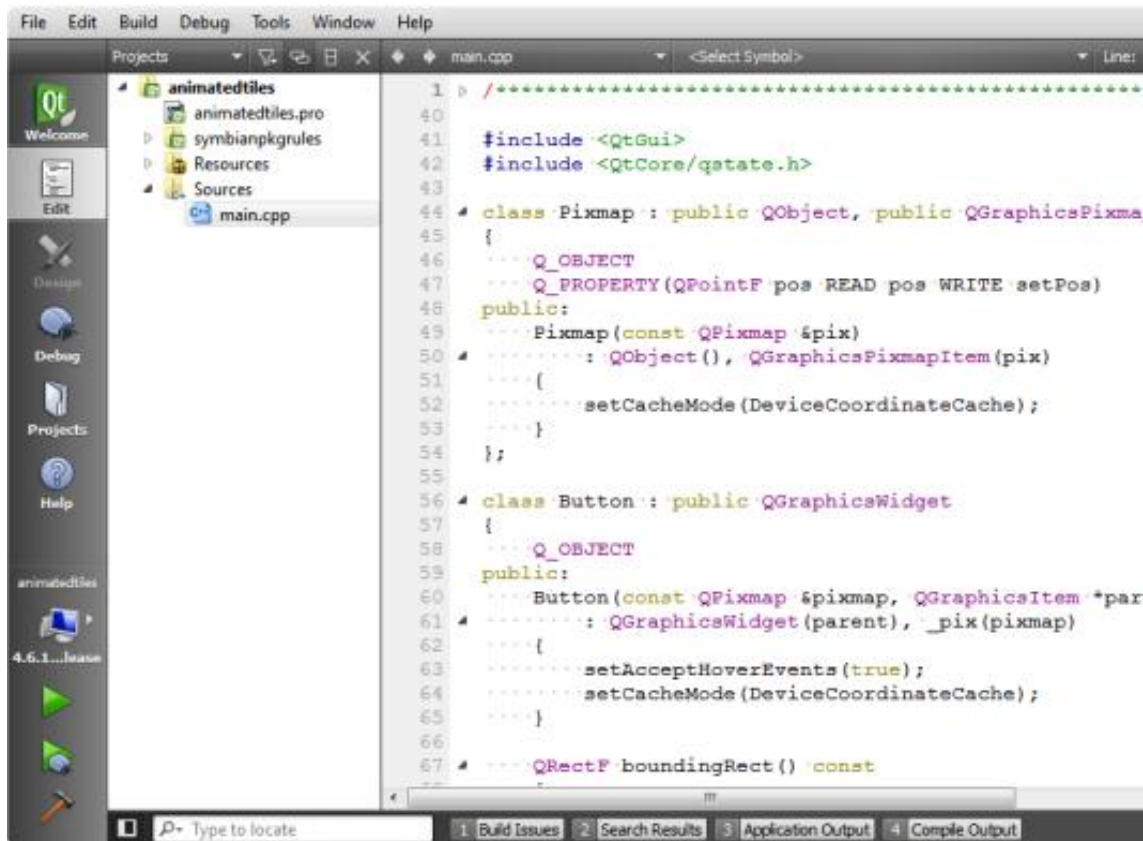
Vuonna 2010 Nokia ja Intel ilmoittivat, että ne yhdistävät Maemon ja Intelin Moblin-alustan yhdeksi mobiilikäyttöjärjestelmäksi, jonka nimeksi tuli MeeGo (Ojanperä 2010). Kesällä 2011 Nokia julkaisi ensimmäisen (ja mahdollisesti viimeisen Nokialta tulevan) MeeGo-laitteen. Laite on Nokia N9 ja se sisältää MeeGo 1.2 -version, joka on varustettu Nokian omalla Harmattan-käyttöliittymällä. (Lehto 2011.)

### **3.2 Menetelmät ja työkalut**

Maemolle ohjelmia voi toteuttaa GTK+:lla, Qt:llä ja Pythonilla. Ensimmäisissä Maemon versioissa Qt ei vielä ollut saatavilla, mutta Nokia julkaisi sen Maemon 5. versioon. Qt:n avulla toteutetut sovellukset toimivat teoriassa monella eri alustalla (Maemo 5, Maemo 6, Symbian...) ilman koodimuutoksia. (Lehtiniitty 2009; Get started with Maemo 5. 2009.)

Sovelluskehitystä varten pitää kehittäjän ladata Maemon internet-sivuilta Maemo SDK, joka sisältää tarvittavat työkalut sovellusten toteuttamista ja testaamista varten. Maemo SDK:n mukana tulevat työkalut toimivat Scratchbox-ympäristössä, joka on käytännössä käyttöjärjestelmän päällä toimiva erillinen Linux-ympäristö, joka toimii kuten Maemo. Sovelluksien testaamista varten SDK:n mukana tulee myös Xephyr-emulaattori, joka näyttää ja toimii kuten oikea Maemo-laite. (Maemo.org. 2011, linkit Intro -> Developing on Maemo.)

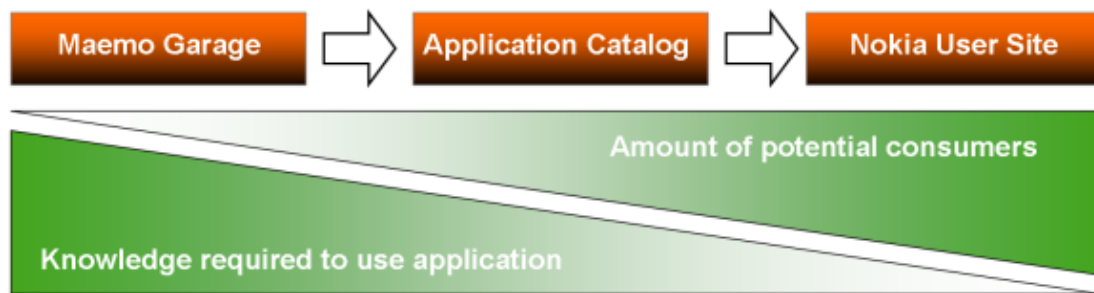
Toteutettaessa sovelluksia Qt:n avulla pitää kehittäjän asentaa vielä erikseen Qt-kirjastot. Tätä varten on Nokian sivuilta ladattavissa Nokia Qt SDK, joka sisältää tarvittavat kirjastot ja työkalut. SDK:n mukana tulee Qt Creator -sovelluskehitysympäristö ja emulaattorit usealle Nokian eri matkapuhelinalustalle (mukana myös Maemo). Qt Creatorin käyttöliittymä on esitetty kuvassa 2. (Qt – Tools. 2011.)



KUVA 2. Qt Creator

### 3.3 Julkaisu

Maemo-sovelluksien julkaisu, alkaen ensimmäisestä prototyypistä aina loppuun asti hiottuun suurelle käyttäjämäärälle tarkoitettuun sovellukseen, on tavallisesti kolmivaiheinen prosessi. Ensimmäisessä vaiheessa sovelluksen ollessa vielä prototyyppi-asteella voi projektin julkaista Maemo Garage -sivustolla. Toisessa vaiheessa, kun sovellus on jo toimiva ja siitä on tehty valmis asennettava paketti, sen voi julkaista Maemo Application Catalogeihin (joita ovat extras, extras-testing ja extras-devel). Kolmannessa vaiheessa, kun käyttäjät ja testaajat ovat todenneet sovelluksen toimivaksi, sen voi julkaista Nokian Ovi Storeen. Kuvassa 3 on esitetty Maemo-sovelluksen julkaisuvaiheet suhteutettuna käyttäjämäärään ja ohjelman asentamiseen tarvittavaan osaamiseen. (Maemo.org. 2011, linkit Intro -> Developing On Maemo.)



KUVA 3. Maemo-sovelluksen julkaisuprosessi (Maemo.org 2011, linkit Intro -> Developing On Maemo)

### Maemo Garage

Maemo Garage on internet-sivusto, jossa kehittäjät voivat julkaista sovellusprojekteja. Kuka tahansa voi liittyä Maemo Garageen ja liittyä johonkin meneillään olevaan projektiin tai aloittaa kokonaan uuden projektin. (Maemo.org. 2011, linkit Intro -> Development Garage.)

Jokaisesta Garagessa julkaistusta sovelluksesta löytyy seuraavat tiedot:

- yleistä tietoa projektista
- projektin tekijät
- projektin tila
- projektin raportoidut viat
- postituslista
- lähdekoodi (Maemo.org. 2011, linkit Intro -> Development Garage).

### Application Catalog

Maemo Application Catalogeilla tarkoitetaan Maemo Extras -sovellusvarastoja (repository), joita ovat extras, extras-testing ja extras-devel. Näiden varastojen kautta käyttäjä voi helposti asentaa sovelluksia laitteeseen. Extras-testing ja extras-devel sisältävät sovelluksia, jotka eivät välttämättä vielä ole täysin valmiita, ja ne on näin ollen tarkoitettu vain kokeneiden käyttäjien testattaviksi. (Maemo.org. 2011, linkit Development -> Upload projects to the Extras repository.)

Kun sovelluskehittäjä saa sovelluksen mielestään valmiiksi julkaistavaksi, hän voi julkaista sen extras-devel-varastoon ja sitä kautta kokeneet käyttäjät voivat testata sovellusta ja antaa palautetta sen toiminnasta. Kun sovellus on extras-devel-varastossa todettu toimivaksi, voi kehittäjä pyytää sen siirtoa extras-testing-varastoon. Extras-testing-varastossa käyttäjät kokeilevat sovellusta ja antavat sille pisteitä sen toiminnan mukaan. Kun pisteitä tulee riittävästi, voi pyytää sovelluksen siirtoa extras-varastoon, joka sisältää sovelluksia, jotka ovat valmiita kaikkien käyttäjien käytettäväksi. (Maemo.org. 2011, linkit Development -> Upload projects to the Extras repository.)

## **Ovi Store**

Ovi Store on Nokian globaali markkinapaikka. Sen kautta miljoonat käyttäjät lähes ympäri maailmaa lataavat sovelluksia, pelejä ja mediaa. Ovi Storesta voi ladata tai ostaa sovelluksia kaikille Nokian eri laitteille. (Ovi Publisher Guide. 2011, 4.)

Julkaistakseen sovelluksen Ovi Storessa pitää julkaisijan rekisteröityä ensin Ovi Storeen. Rekisteröintivaiheessa julkaisijan pitää maksaa yhden euron (1 €) suuruinen rekisteröitymismaksu, jonka jälkeen hän voi julkaista sovelluksia Ovi Storeen. (Ovi Publisher Guide. 2011, 7–15.)

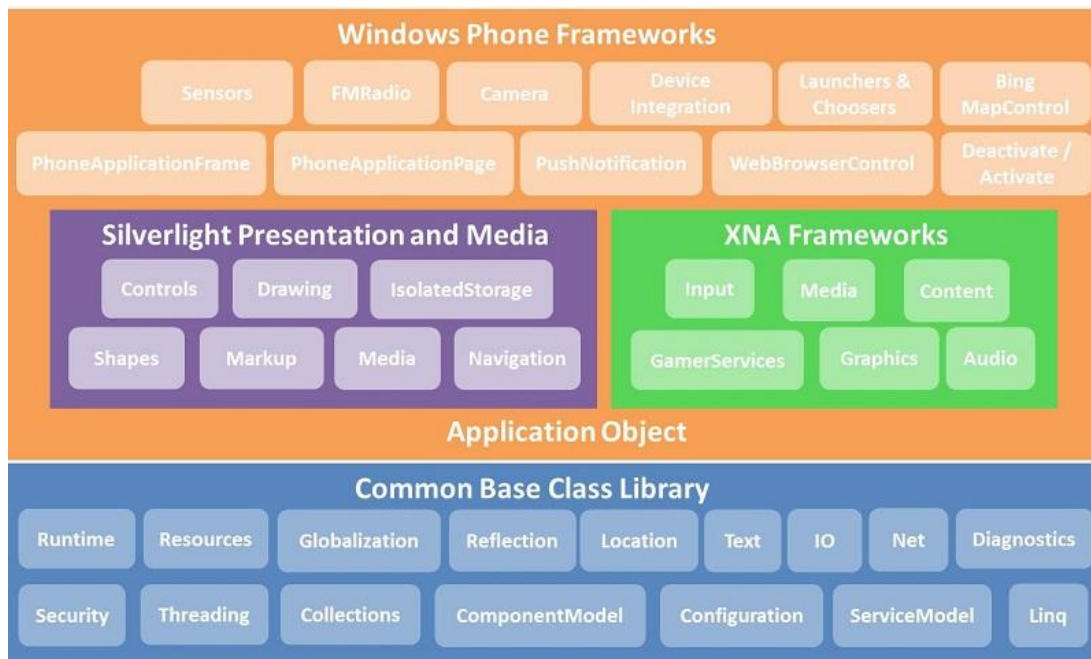


## **4 OHJELMISTOKEHITYS WINDOWS PHONE 7:LLE**

Tässä luvussa esitellään Windows Phone 7 -mobiilikäyttöjärjestelmä, menetelmät ja työkalut sovellusten toteuttamiseen sille sekä Windows Phone 7 -sovellusten julkaisuprosessi.

### **4.1 Windows Phone 7 -käyttöjärjestelmä**

Windows Phone 7 -käyttöjärjestelmä on Microsoftin kehittämä mobiilikäyttöjärjestelmä, jossa on pyritty yhdistämään hyviä ominaisuuksia, joihin käyttäjät ovat tottuneet Applen iPhonessa ja Android-pohjaisissa puhelimissa. Näitä ominaisuuksia ovat muun muassa monikosketus (multitouch), sosiaaliset verkostopalvelut (kuten Facebook) ja tuki yleisimmille sähköpostitileille (Yahoo, Hotmail, Google, AOL, Microsoft Exchange). Lisäksi Microsoft toimittaa käyttöjärjestelmän mukana Office-version, jolla voi lukea, muokata, tallentaa ja synkronoida Word-, Excel- ja muita Office-dokumentteja. Microsoft loi käyttöjärjestelmää varten kokonaan uudenkaltaisen, modernia designia sisältävän käyttöliittymän, jonka se nimesin Metroksi. Windows Phone 7 -alustan komponentit on esitetty kuvassa 4. (Lee – Chuvyrov 2010, 5.)



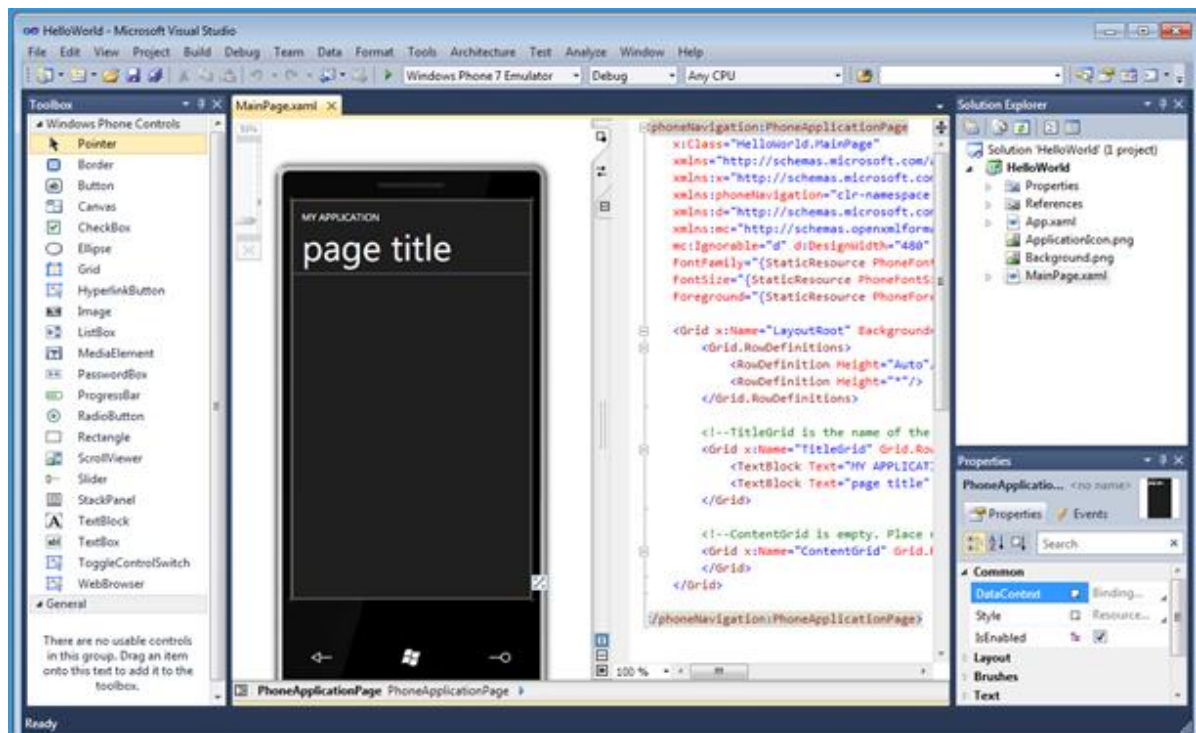
KUVA 4. Windows Phone 7 -alustan komponentit (Application Platform Overview for Windows Phone. 2011)

## 4.2 Menetelmät ja työkalut

Microsoft ei kehittänyt uusia menetelmiä ja työkaluja ohjelmistojen kehitykseen Windows Phone 7 -alustalle, vaan se mukautti olemassa olevia menetelmiään Windows Phone 7:lle sopivaksi. Tämä tarkoittaa sitä, että WP7:lle voi toteuttaa ohjelmia C#-ohjelmointikielen ja .NET-kehiksen avulla. .NET-kehiksen päällä toimii kaksi erilaista kehystä, Silverlight ja XNA, joiden avulla kehittäjä voi toteuttaa varsinaisen sovelluksen. Silverlight on tarkoitettu yrityssovellusten sekä yksinkertaisten 2D-pelien toteutukseen ja XNA on tarkoitettu monimutkaisempien pelien tekemiseen. (Lee – Chuvrov 2010, 7–12.)

Kehittäjät voivat ladata ja asentaa Microsoftin sivuilta ilmaiseksi paketin, joka sisältää kaikki työkalut, jotka tarvitaan sovellusten tekemiseen Windows Phonelle. Pakettiin kuuluvat Visual Studio 2010 Express for Windows Phone, Expression Blend, Windows Phone Emulator ja XNA Game Studio. (Application Platform Overview for Windows Phone. 2011.)

Visual Studio 2010 on sovelluskehitysympäristö, jonka avulla voidaan toteuttaa Windows Phone -sovelluksia Silverlightin tai XNA:n avulla. Visual Studio sisältää designerin, debuggerin, projektijärjestelmän, paketoijan ja julkaisugeneraattorin. Kuvassa 5 on esitetty Visual Studion käyttöliittymä. (Lee – Chuvyrov 2010, 7–12.)



KUVA 5. Visual Studio 2010 Express for Windows Phone

Expression Blend on työkalu, jolla voidaan toteuttaa monipuolisempia graafisia XAML-pohjaisia käyttöliittymiä. Expression Blendin avulla tehdyt käyttöliittymän osat voi siirtää Visual Studioon ja toteuttaa niiden toiminnallisuudet Visual Studion avulla. (Lee – Chuvyrov 2010, 7–12.)

XNA Game Studio on työkalu, jonka avulla voidaan toteuttaa pelejä Windowsille, Xbox 360:lle, Zunele ja Windows Phonelle. XNA Game Studio laajentaa Visual Studio -työkaluja tukemaan XNA-kehystä. (Application Platform Overview for Windows Phone. 2011.)

### 4.3 Julkaisu

Windows Phone 7:lle sovelluksia julkaistaan Microsoftin Marketplacen kautta. Sovellusten julkaisua varten kehittäjän pitää rekisteröityä Microsoft App Hub -palveluun, jonka kautta sovellukset julkaistaan Marketplaceen. Myös testatakseen sovellusta laitteessa kehittäjän pitää ensin rekisteröityä App Hubiin ja rekisteröidä testikäytössä oleva puhelin testilaitteeksi. (Lee – Chuvyrov 2010, 12–14.)

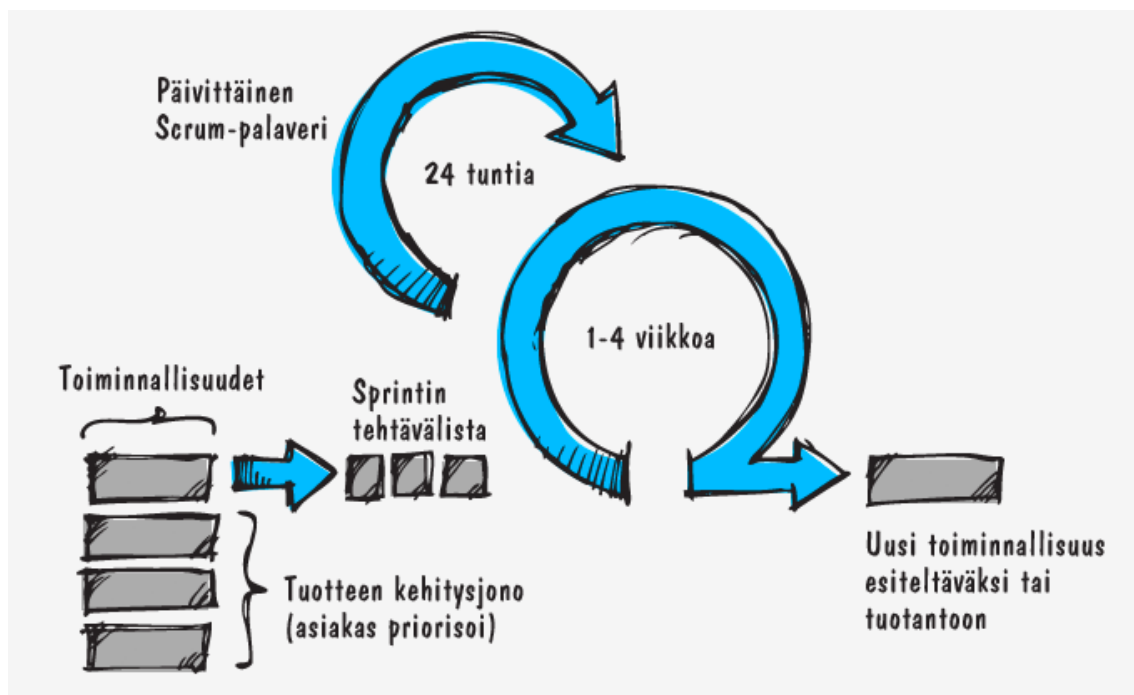
App Hubin käyttöä varten pitää kehittäjän maksaa rekisteröitymisvaiheessa vuosimaksu, joka on 99 dollaria (\$99). Poikkeuksena ovat opiskelijat, jotka saavat käyttää palvelua ilmaiseksi vahvistamalla opiskelunsa DreamSpark-palvelun kautta. Rekisteröitymisen jälkeen pitää kehittäjän vielä vahvistaa henkilöllisyytensä. Tätä varten GeoTrust-niminen Microsoftin yhteistyökumppani ottaa yhteyttä ja pyytää lähettämään kuvan passista tai ajokortista. (App hub registration walkthrough. 2011.)

Kun kehittäjä saa sovelluksensa valmiiksi, hän voi siirtää sen App Hubiin, josta se siirtyy automaattisesti sertifiointiprosessiin, joka varmistaa, että sovellus täyttää kaikki Microsoftin asettamat vaatimukset. Kun sovellus on läpäissyt sertifiointin, se siirtyy Marketplaceen käyttäjien ladattavaksi. (Lee – Chuvyrov 2010, 12–14.)

## 5 SCRUM – KETTERÄ KEHITYSMENETELMÄ

### 5.1 Scrum-prosessi

Scrum on ketterä ohjelmistokehitysmenetelmä. Scrumia voidaan käyttää erisuuruisissa ohjelmistoprojekteissa, mutta se soveltuu erityisen hyvin komplekseihin projekteihin. Scrumissa projektille asetetut vaatimukset eivät aluksi ole tarkkoja, vaan ne elävät ja tarkentuvat projektin edetessä. Scrum mahdollistaa projektin etenemisen tarkan seurannan ja mahdollisiin hidastaviin tekijöihin voidaan puuttua jo varhaisessa vaiheessa. Scrum-prosessi on esitetty kuvassa 6. (Lindström 2011.)



KUVA 6. Scrum-prosessi (Lindström 2011)

Scrumia käytettäessä sovelluskehitysprosessi jaetaan lyhyisiin 1–4 viikon pituisiin sprintteihin. Jokaisen sprintin jälkeen pidetään sprintin katselmointi, jossa tarkastellaan, mitä on saatu tehtyä, ja päätetään seuraavassa sprintissä tehtävät asiat. Sprintin aikana pidetään joka työpäivä päivittäinen seurantapalaveri, päivän Scrum (Daily Scrum), jossa jokainen tiimin jäsen kertoo, mitä on saanut tehtyä ja mitä aikoo tehdä seuraavaksi. Näin saadaan

projekti pysymään hyvin aikataulussa ja mahdolliset ongelmat havaitaan jo varhaisessa vaiheessa. (Lindström 2011.)

## **5.2 Roolit**

Perinteisessä ohjelmistoprojektissa on yleensä määrittelijöitä, suunnittelijoita, ohjelmoijia, testaajia ja projektipäällikkö. Scrum-projektissa sen sijaan on vain kolme eri roolia, jotka ovat tuotteen omistaja, Scrum-mestari ja tiimi. (Ketterät käytännöt.fi. 2008.)

### **Tuotteen omistaja**

Tuotteen omistaja (product owner) ohjaa koko kehitysprosessia liiketoiminnan ja asiakkaan tarpeiden näkökulmasta. Asiakasprojekteissa tuotteen omistaja voi olla asiakkaan edustaja. Tuotteen omistaja päättää kaikki tuotteen ominaisuudet ja toiminnallisuudet. Projektin onnistumisen kannalta tuotteen omistajan täytyy olla tiiviissä yhteistyössä koko Scrum-tiimin kanssa. (Lindström 2011.)

### **Scrum-mestari**

Scrum-mestari (Scrum master) vastaa koko Scrum-prosessin etenemisestä. Scrum-mestarin tehtävä on pitää huoli, että koko tiimi voi tehdä töitä mahdollisimman optimaalisella tavalla. Scrum-mestarin tehtävänä on myös ratkoa tiimiläisten ilmoittamia työtä hidastavia ongelmia ja huolehtia, että koko tiimi noudattaa Scrumia oikein. (Ketterät käytännöt.fi. 2008.)

### **Tiimi**

Tiimiin (Scrum team) kuuluvat kaikki henkilöt, jotka projektiin osallistuvat. Tiimin sisällä ei ole erikseen määritelty tarkkaa tehtävää jokaiselle tiimiläiselle, vaan tiimiin kootaan henkilöitä, joilta löytyy tarvittava osaaminen kaikkiin projektiin liittyviin tehtäviin. Tällä tavalla voidaan korostaa jokaisen tiimiläisen olevan tasavertaisessa asemassa projektin valmistumisen kannalta. Koko tiimi on

vastuussa projektin etenemisestä, ja tiimin sisällä tehtävät jaetaan yhteisöllisesti muiden tiimiläisten kesken osaamisen mukaan eikä pompotella tehtäviä tekijältä toiselle. (Ketterät käytännöt.fi. 2008.)

## **5.3 Dokumentit**

Tärkeimpiä dokumentteja Scrumin käytössä ovat tuotteen kehitysjono, sprintin tehtävälista ja julkaisun edistymiskäyrä (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project?).

### **Tuotteen kehitysjono**

Tuotteen kehitysjono (product backlog) on lista kaikista tuotteeseen tulevista toiminnallisuuksista. Kehitysjonon priorisoi tuotteen omistaja, jotta tiimi voi aina työskennellä tärkeimpien toiminnallisuuksien parissa. (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> product backlog.)

Projektin alkuvaiheessa tuotteen kehitysjonoon kirjataan ylös kaikki mieleen tulevat tuotteeseen halutut toiminnallisuudet. Listan kokoaa Scrum-tiimi yhdessä tuotteen omistajan kanssa. Tämän listan pohjalta päästään työssä alkuun, ainakin ensimmäisen sprintin ajan, ja sen jälkeen ennen seuraavaa sprinttiä tuotteen omistaja toimittaa tiimille listan, joka sisältää kaikki aiemmin mietityt toiminnallisuudet, mutta ne on priorisoitu niiden tärkeyden mukaan. Tämän jälkeen tiimi valitsee listalta korkeimmin priorisoituja kohteita, jotka jäsenet arvioivat voivansa toteuttaa seuraavan sprintin aikana, ja siirtää ne sprintin tehtävälistaan. Lisäksi tiimi voi valita sprintin aikana toteutettavaksi myös matalamman prioriteetin kohteita, jos korkeamman prioriteetin kohteiden toteutuksen jälkeen jää vielä ylimääräistä aikaa. Kuvassa 7 on esitetty esimerkki priorisoidusta tuotteen kehitysjonosta. (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> product backlog.)

	Item #	Description	Est	By
<b>Very High</b>				
	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
		- Add licensing	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		<b>Analysis Manager</b>		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
<b>High</b>				
		- Enforce unique names	-	-
	7	In main application	24	KH
	8	In import	24	AM
		- Admin Program	-	-
	9	Delete users	4	JM
		- Analysis Manager	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
		- Query	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
		- Population Genetics	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	Add icons for v1.1 or 2.0	-	-
		- Pedigree Manager	-	-
	20	Validate Derived kindred	4	KH
<b>Medium</b>				
		- Explorer	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

KUVA 7. Esimerkki tuotteen kehitysjonosta (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> product backlog)

Tuotteen kehitysjonoon kirjataan myös jokaiselle vaatimukselle arvio aikamäärästä, joka sen toteuttamiseen tiimin mielestä kuluu. Tuotteen kehitysjono elää koko projektin ajan sitä mukaa, kun toiminnallisuuksia saadaan toteutettua ja vaatimukset tarkentuvat sekä tulee uusia vaatimuksia. (Schwaber – Sutherland 2011.)

### Sprintin tehtävälista

Sprintin tehtävälista (sprint backlog) on lista tehtävistä, jotka tiimin pitää toteuttaa sprintin aikana toteutettavaksi valitun toiminnallisuuden tavoittamiseksi. Tehtävälistan kokoamisessa on tärkeää, että siihen osallistuu koko Scrum-tiimi, koska ainoastaan tiimi voi arvioida toiminnallisuuksien



toteuttamiseen kuluvan ajan. (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> sprint backlog.)

Sprintin tehtävälista on tyypillisesti Excel-taulukko, mutta sen luomiseen ja koko Scrum-prosessin hallintaan on saatavilla myös valmiita ohjelmistoja. Kuvassa 8 on esitettyä hyvin yksinkertainen esimerkki sprintin tehtävälustasta. (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> sprint backlog.)

Tasks	Mon	Tues	Wed	Thurs	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

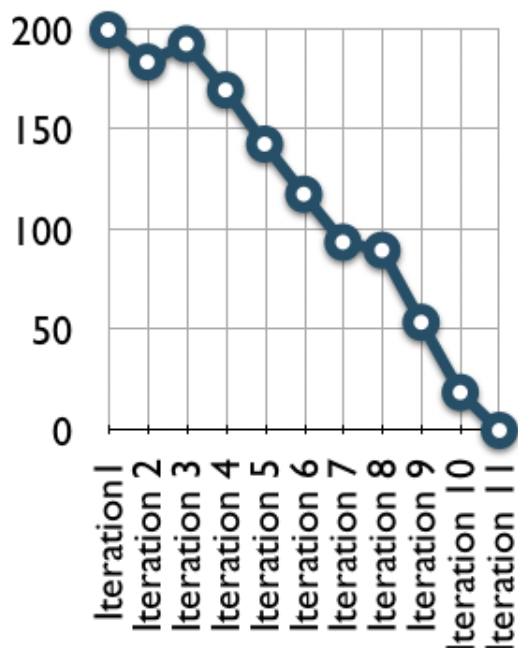
KUVA 8. Esimerkki sprintin tehtävälustasta (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> sprint backlog)

Sprintin tehtävälustaa ylläpitää Scrum-mestari, joka päivittää tehtävälustaan suoritettut tehtävät ja arvion tarvittavasta ajasta, joka kuuluu vielä toteuttamattomien tehtävien toteuttamiseen. Näiden arvioiden avulla voidaan toteuttaa myös sprintille oma sprintin edistymiskäyrä samaan tapaan kuin julkaisun edistymiskäyrä (ks. Julkaisun edistymiskäyrä). Tämän käyrän avulla voidaan seurata, eteneekö sprintti aikataulussa. Tiimi pyrkii parhaansa mukaan arvioimaan tehtäviin kuluvaan aikaan, mutta joskus arviot ovat joko liian pieniä tai liian suuria. Jos tarvittava aika on arvioitu liian pieneksi, sen seurauksena sprintti ei etene aikataulussa ja siinä tapauksessa tiimi joutuu jättämään joitain toteutettavaksi valittuja tehtäviä toteuttamatta. (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> sprint backlog.)

## Julkaisun edistymiskäyrä

Julkaisun edistymiskäyrä (release burn-up/burn-down) kertoo jäljellä olevan työmäärän ennen tuotteen julkaisua. Julkaisun edistymiskäyrä on tehokas tapa seurata projektin etenemistä ja varmistaa, että projekti valmistuu sovitussa ajassa. (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> release burndown chart.)

Julkaisun edistymiskäyrässä yleensä verrataan tehtyä työmäärää jäljellä oleviin sprintteihin. Työmäärän yksikkö on vapaasti tiimin päätettävissä ja kokonaistyömäärä lasketaan tuotteen kehitysjonoon merkityistä arvioituista työmääristä. Julkaisun edistymiskäyrä voi olla tyypiltään joko laskeva (burn-down) tai nouseva (burn-up). Kuvassa 9 on esimerkki laskevasta edistymiskäyrästä. (Schwaber – Sutherland 2011.)



KUVA 9. Esimerkki julkaisun edistymiskäyrästä (Mountain Goat Software 2011, linkit Scrum -> What are the main artifacts of a Scrum Project? -> release burndown chart)

## **6 TOTEUTUS**

Tässä luvussa kuvataan musiikkisoittimen sovelluskehitysprosessi molemmille alustoille. Työn suorittamisen aikana toteutettiin ensin musiikkisoitin Maemolle ja sen jälkeen Windows Phone 7:lle, ja siksi niiden kehitysprosessi on kuvattu toisistaan erillään.

### **6.1 Vaatimusmäärittely**

Vaatimusmäärittelyssä kirjattiin ylös sovellukselle määrättyjä vaatimuksia. Koska käytössä oli Scrum, vaatimukset kirjattiin tuotteen kehitysjonoon ja ne elivät työn toteutuksen aikana sitä mukaa, kun vaatimukset tarkentuivat. Koska projektin toteuttamisen aikana vasta tutustuttiin Scrumiin, jätettiin vaatimuksien toteuttamiseen kuluvan ajan arviointi pois ja sitä kautta myös julkaisun edistymiskäyrää ei projektille tehty. Taulukossa 1 näkyy, miltä toteutettavien sovellusten tuotteen kehitysjono työn loppuvaiheessa näytti.

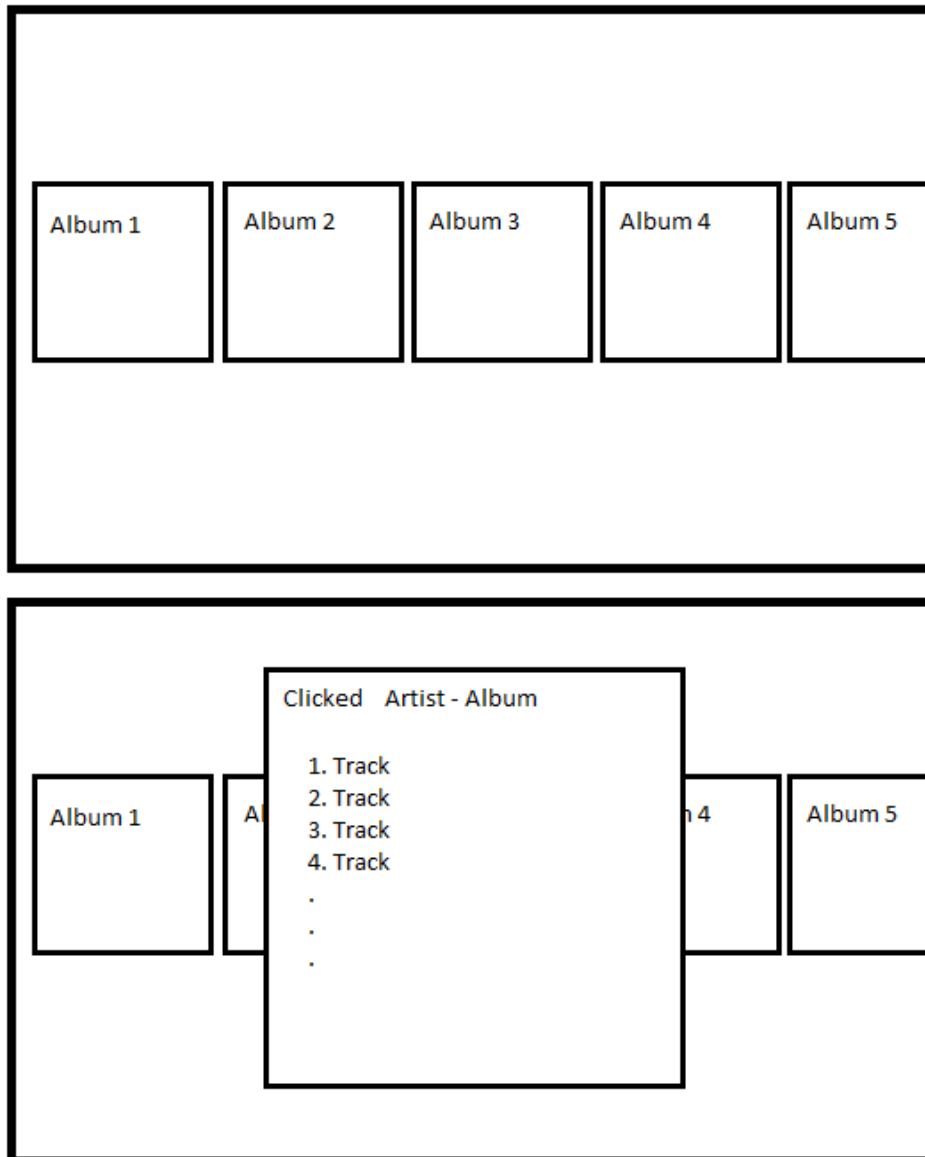
### QMLPlayer/WP7-player product backlog

#	Feature	Status	Sprint(s)
Very high			
1	Starting projects	Done	1,3,4
2	QMLPlayer general	In progress	1,2,3
3	WP7 player general	In progress	4,5,6
5	Music Playback (Maemo)	Done	2,3
9	Album Art Clicking (Maemo)	Done	2,3
13	Album Art UI (WP7)	Done	5,6
14	Music Playback (WP7)	Done	5
High			
4	Album Art Flow (Maemo)	Done	1,2,3
10	Get data from Tracker (Maemo)	Done	2,3
6	Track list view (Maemo)	Done	2,3
11	Play/Pause music actions (Maemo)	Done	2,3
12	Get music information from "system" (WP7)	Done	5,6
15	TrackListView (WP7)	Done	6,7
Medium			
7	Menu for Track objects (Maemo)	Canceled	2,3
8	Next action (Maemo)	Done	2,3
Low			
12	Get Album Arts from internet (optional)	Not Started	3

TAULUKKO 1. Tuotteen kehitysjonon näkymä työn loppuvaiheessa

## 6.2 Suunnittelu

Suunnitteluvaiheessa suunniteltiin sovelluksen toteutusta, kuten käytettävät tekniikat, arkkitehtuuri ja käyttöliittymä. Tavoitteena oli toteuttaa vastaavat soittimet molemmille alustoille, ja sen vuoksi suunnitteluvaiheessa syntyi vain karkea käyttöliittymäluonnos, jonka pohjalta lähdettiin sovellukset toteuttamaan molemmille alustoille pienin muutoksin. Karkea käyttöliittymäluonnos on esitetty kuvassa 10.



KUVA 10. Suunniteltu käyttöliittymä

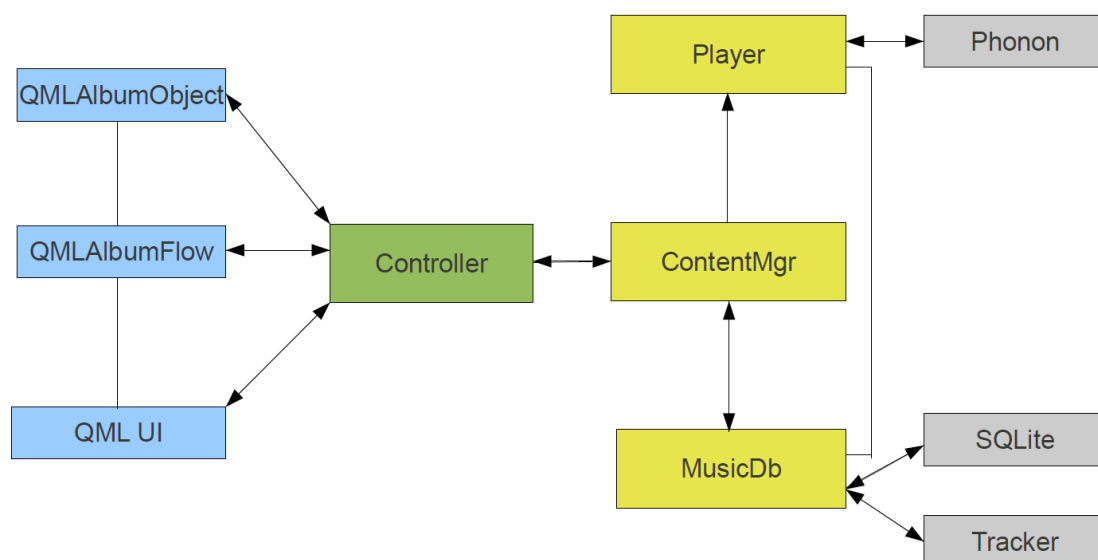
## 6.3 Maemo 5

### 6.3.1 Arkkitehtuuri

Arkkitehtuurikuvassa (kuva 11) on kuvattuna kaikki sovellukseen suunnitellut osat ja niiden kommunikointi keskenään. Kuvan oikeassa laidassa näkyy, miten sovellus hyödyntää Maemossa olevaa Phonon-kirjastoa musiikin toistoon, SQLite-kirjastoa musiikkitietokannan tekemiseen sekä Tracker-sovellusta musiikkitietojen kyselyyn. Tracker-sovellukselta saadaan tieto kaikista laitteessa

olevista musiikkitiedoista ja ne tallennetaan omaan SQLite-tietokantaan. Tätä kautta sovelluksen toimintaa saadaan hieman nopeutettua, kun ei joka kerta tarvitse hakea tietoja uudelleen Trackerilta.

Kuvan vasemmassa laidassa sinisellä pohjalla olevat osat ovat käyttöliittymän osia, jotka toteutettiin QML:n avulla, ja ne kommunikoivat Controller-luokan kanssa. Controller-luokka välittää käyttöliittymäkomponenteilta tulevat komennot eteenpäin ContentMgr-luokalle ja toisinpäin. MusicDb-luokka käsittelee musiikkitietokantaa ja Player-luokka hoitaa musiikin toistoon liittyvät toimenpiteet ja kommunikoi Phononin kanssa.



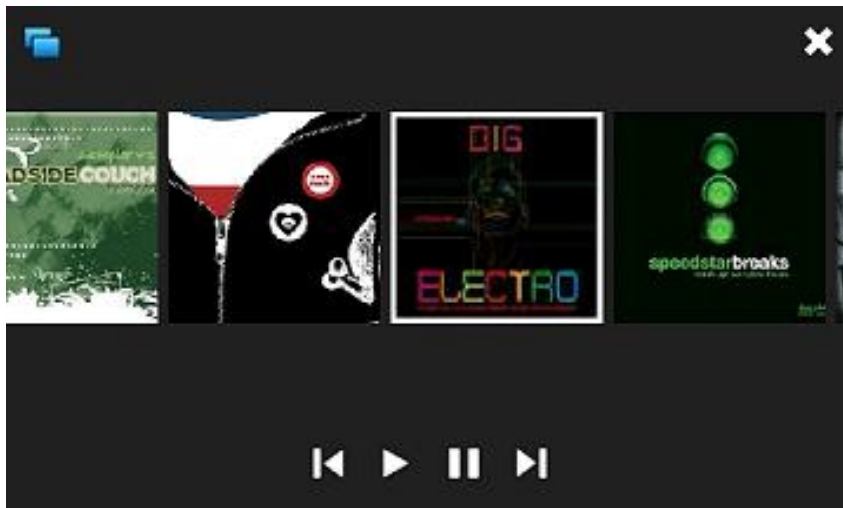
KUVA 11. Musiikkisoittimen suunniteltu arkkitehtuuri

### 6.3.2 Toteutus

Sovellukseen toteutettiin alkuperäisen suunnitelman mukaisesti käyttöliittymä QML:n avulla ja toiminnallisuudet Qt:llä käyttäen ohjelmointikielenä C++:aa. Koska aiemmissa harjoitteluprojekteissa oli toteutettu myös musiikkisoitin, joka sisälsi osittain samoja toiminnallisuuksia, sai sen lähdekoodista hyviä malleja sovelluksen toteutukseen. Sovellus toteutettiin siten, että se hakee jokaisen käynnistyserran yhteydessä Trackerilta tiedot laitteesta löytyvistä musiikkitiedostoista. Sovelluksen käynnistymisen kannalta tämä ei ole paras

mahdollinen ratkaisu ja vaatisi hieman jatkokehitystä (ladataan tiedot vain kun uusia kappaleita on lisätty). Ajanpuutteen vuoksi tätä ei kuitenkaan enää lähdetty kehittämään.

Sovelluksen käyttöliittymän päänäkömään tuli näkymään kaikkien laitteesta löytyvien albumien kuva, jos kuvaa ei löytynyt, artistin ja albumin nimi (kuva 12). Albumin kuvaa näpäyttäessä se suurentuu isommaksi ja samalla albumin toisto alkaa ensimmäisestä kappaleesta (kuva 13). Kun avoinna olevaa albumia näpäytetään uudestaan, se ”kääntyy” ympäri ja tulee näkyviin kappalenäkymä, jossa on listattuna kaikki albumilta löytyvät kappaleet. Kappaleita näpäyttämällä voidaan toistaa yksittäisiä kappaleita (kuva 14).



KUVA 12. Käyttöliittymän päänäkömää



KUVA 13. Käyttöliittymän näkymä, kun albumia näpäytetty



KUVA 14. Käyttöliittymän näkymä, kun avointa albumia näpäytetty

Toteutusvaiheessa toteutettiin myös Debian-paketointia varten tarvittavat tiedot sovellukselle. Julkaisua varten täytyy sovelluksesta tehdä Debian-paketti, joka sisältää tiedot sovelluksen riippuvuuksista, ikonit, asennushakemistot ja kaiken muun tarpeellisen sovelluksen asentamiseen laitteelle.

### 6.3.3 Testaus

Sovelluksen testaus toteutettiin lähes pelkästään toiminnallisena testauksena suoraan kohdelaitteessa (Nokia N900). Samalla testatessa seurattiin, kuinka paljon sovellus kuormittaa laitteen suoritinta, ja verrattiin kuormitusta laitteen



alkuperäiseen musiikkisoittimeen. Sovellus käytti suoritinta ja muistia hieman enemmän kuin alkuperäinen. Tämän arvioitiin johtuvan siitä, että toteutettu soitin sisältää enemmän graafisia ominaisuuksia kuin alkuperäinen soitin.

Sovellusta testatessa kokeiltiin mahdollisia eri käyttötapauksia ja tarkkailtiin, käyttäytyykö sovellus halutulla tavalla. Joitain pieniä vikoja sovelluksesta löytyikin, mutta niistä suurin osa saatiin suosiolla korjattua.

#### **6.3.4 Julkaisu**

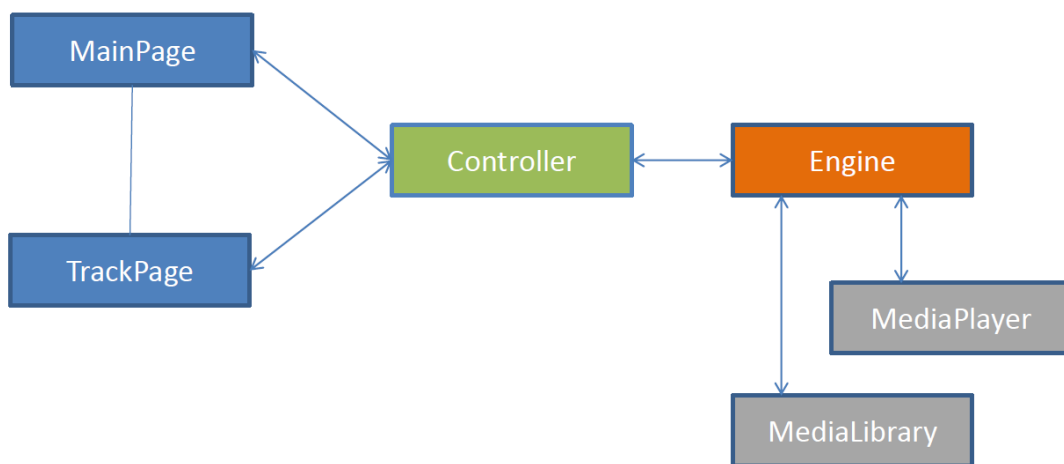
Sovelluksen julkaiseminen olisi onnistunut Maemo extras-devel repositoryyn, mutta sovelluksen hitaan käynnistymisen vuoksi (selitys kappaleessa 7.3.2) sitä ei julkaistu. Sovelluksesta toteutettu Debian-paketti on kuitenkin sellainen, että se voidaan asentaa laitteelle ilman julkista välityskanavaa. YouTubeen ladatun videon kommenttien perusteella soitin herätti kiinnostusta ja käyttäjiä olisi voinut löytyä. Todennäköisesti Maemo 5 -laitteiden määrä kuitenkin koko ajan vähenee uusien laitteiden myötä ja sen vuoksi ei soittimen jatkokehittämiselle ja julkaisemiselle enää riittänyt innostusta. Tekstin kirjoitushetkellä Nokialta oli tulossa markkinoille MeeGo-käyttöjärjestelmällä varustettu matkapuhelin (Nokia N9), jolle sovellus olisi suhteellisen helposti muutettavissa ja sitä kautta käyttäjiä voisi olla myös tulevaisuudessa.

### **6.4 Windows Phone 7**

#### **6.4.1 Arkkitehtuuri**

Windows Phone 7:lle soitinta suunniteltaessa tuli luettua runsaasti dokumentaatioita ja tutkittua, minkälaisia mahdollisuuksia alusta tarjoaa haluttuun tarkoitukseen. Jo suunnitteluvaiheessa kävi selväksi, että WP7:lle soittimen toiminnallisuuksien toteuttaminen tulee olemaan paljon helpompaa kuin Maemolle, koska WP7:sta löytyi suoraan MediaLibrary-kirjasto, jolta saa helposti tiedot kaikista laitteen musiikkitiedostoista.

Sovellukseen suunniteltiin toteutettavaksi Controller- ja Engine-luokat. Controller-luokka välittää käyttöliittymältä tulevat komennot Enginelle ja toisinpäin. Engine hoitaa kommunikoinnin MediaLibrary- ja MediaPlayer-luokkien kanssa. Kuvassa 15 on esitetty suunniteltu arkkitehtuuri.



KUVA 15. Musiikkisoittimen suunniteltu arkkitehtuuri

## 6.4.2 Toteutus

Alkuperäinen suunnitelma oli toteuttaa sovellus WP7:lle pelkän Silverlightin avulla, mutta työn tekemisen aikana tuli ilmi, että musiikkitiedot sisältävä MediaLibrary-luokka löytyy vain XNA-Frameworkista. Lopulta sovellus tuli siis käyttämään sekä Silverlightia että XNA:ta.

Toteutusvaiheessa tuli ilmi, että WP7 tarjoaa kuvien käsittelyyn vain rajoitetut toiminnallisuudet, ja sen vuoksi käyttöliittymää jouduttiin vähän yksinkertaistamaan suunnitellusta. Sovelluksen päänäkömään toteutettiin vain lista, jossa näkyvät kaikkien albumien kuvat, artistin nimi ja albumin nimi. Albumia näpäyttämällä avautui sitten lista, jossa näkyy kaikki albumilla olevat kappaleet. Kuvassa 16 näkyy sovelluksen toteutunut käyttöliittymä.



KUVA 16. Vasemmalla soittimen päänäköymä, oikealla kappalenäköymä

### 6.4.3 Testaus

Sovelluksen testaus suoritettiin kuten Maemollakin. Tässä vaiheessa kuitenkin tuli eteen eräs työvaihe, jota Maemolla ei tarvittu. Windows Phone 7:lle piti tehdä developer unlock, jos sillä halusi testata tehtyä sovellusta käytännössä. Tätä varten piti ensin rekisteröityä Microsoft App Hub -palveluun ja suorittaa luodun tunnuksen vahvistus. Vahvistusta varten piti odottaa yhteydenottoa GeoTrust-nimiseltä yritykseltä, joka vaati lähettämään kopion passista tai ajokortista. Tunnuksen aktivointia joutui rekisteröinnin jälkeen siis kokonaisuudessaan odottelemaan noin viikon ja vasta sen jälkeen onnistui sovelluksen testaaminen puhelimessa.

#### **6.4.4 Julkaisu**

Sovelluksen julkaiseminen olisi onnistunut helposti Microsoft App Hubin kautta, koska aiemmassa vaiheessa piti jo rekisteröityä sinne. Visual Studiolla olisi saanut helposti tehtyä tiedoston, joka olisi suoraan voitu ladata App Hub -palveluun. Lisäksi olisi pitänyt tehdä vain sovellukselle ikonit ja ottaa kuvankaappaukset sovelluksesta. Tässä kohtaa tuli vastaan toinen WP7:n rajoitus, koska sillä ei voi suoraan ottaa kuvankaappausta. Sovellusta ei lopulta julkaistu App Hubilla ollenkaan, koska sen ei nähty tuovan mitään varsinaista etua WP7:n alkuperäiseen ja hyvin toimivaan musiikkisoittimeen verrattuna.

## 7 VERTAILU

Tässä luvussa vertailen Maemo 5- ja Windows Phone 7 -alustoja omien kokemuksieni ja mielipiteideni pohjalta.

### 7.1 Kehitystyökalut

Mielestäni sekä Maemo-kehityksessä käytetty Qt Creator että WP7-kehityksessä käytetty Visual Studio 2010 Express ovat hyviä tarkoitukseensa. Ainoa merkittävä puute, jonka kummastakaan keksin sanottavaa, on se, ettei Microsoftin kehitystyökaluja ole saatavilla Linux-käyttöjärjestelmällä varustetuille tietokoneille ollenkaan, vaan kehittäjällä on pakko olla käytössään Windows-tietokone. Qt Creator sen sijaan on saatavilla sekä Windowsille että Linuxille ja Qt-pohjaisia sovelluksia voi myös toteuttaa sekä Windows- että Linux-pohjaisille PC-tietokoneille.

### 7.2 Toteutus

Mielestäni sovelluksen toteutus oli helpompi WP7:lle, koska sen työkaluista löytyi valmiina kirjastot kaikille sovelluksessa tarvittaville asioille. Toisaalta jos jotain haluttua kirjastoa ei työkaluista valmiina löydy, voi sen toteuttaminen itse olla varsin haastavaa. Kokonaisuutena voi WP7:sta sanoa, että sille on helppo toteuttaa yksinkertaisia sovelluksia, mutta rajoitettujen rajapintojen vuoksi monimutkaisempien sovelluksien toteutus voi olla haastavaa. Sovelluksen toteutuksessa Maemolle käytin käytännössä C++:aa ja sille saa asennettua paljon lisäkirjastoja itse. Näin ollen WP7:lle oli helpompi toteuttaa sovellus, mutta se tarjosi vain rajalliset mahdollisuudet. Maemolle sovelluksen toteuttaminen oli hieman työläämpi, mutta sillä mahdollisuudet olivat huomattavasti laajemmat.

## 7.3 Julkaisu

Sovellusten julkaiseminen olisi ollut lopulta helpompaa WP7:lle, koska sovelluksen testaaminen laitteella vaati jo rekisteröitymistä Microsoftin App Hubiin. Rekisteröitynyt käyttäjä olisi voinut julkaista sovelluksen Marketplaceen tekemällä Visual Studiolla paketin toteutetusta sovelluksesta ja lataamalla sen App Hubiin. Lisäksi olisi tarvinnut vain ikonit ja kuvankaappaukset. Sovelluksen julkaisuvaiheessa App Hubin kautta sovellukselle tehdään jonkinlainen testaus, jossa varmistetaan, että sovellus toimii halutulla tavalla, ja sen jälkeen se siirtyy Marketplacen kautta ladattavaksi. Mielestäni kuitenkin App Hubin 99 dollarin vuosimaksu voi jossain määrin vähentää joidenkin kehittäjien halua julkaista sovelluksia WP7:lle.

Maemolla julkaisu vaati Debian-paketin tekemisen sovelluksesta, mikä on kokemattomalle aikaa vievä työvaihe. Paketoinnin jälkeen sovellus olisi ollut helposti julkaistavissa Maemo extras-devel -varastoon ja sitä kautta käyttäjien ladattavaksi. Extras-varastot (extras, extras-testing, extras-devel) eivät kuitenkaan ole oletuksena kaikkien Maemo-käyttäjien laitteessa valmiina, vaan ne pitää erikseen määritellä ja näin ollen potentiaalisten käyttäjien määrä vähenee. Extras-devel-varastosta sovelluksen saisi pyytämällä siirrettyä extras-testing-varastoon, ja jos käyttäjät toteavat sen toimivaksi, se siirtyisi extras-varastoon. Paras tapa sovelluksen julkaisemiselle olisi kuitenkin Nokia Ovi Store, mutta sen käyttö vaatisi rekisteröitymisen ja rekisteröitymismaksun (1 €) maksamisen.

## 8 YHTEENVETO

Työn tekemisen aikana muodostui selkeä kuva sekä Maemo 5- että Windows Phone 7 -alustoista. On erittäin vaikeaa suoranaisesti sanoa, kumpi alusta on selkeästi parempi, koska kummassakin on sekä hyvät että huonot puolensa. Vaikka kokemusta oli aikaisemmin kertynyt pelkästään Maemo-sovellusten kehityksestä, siitä huolimatta onnistuin aika nopeasti opettelemaan myös WP7-sovellusten kehittämiseen käytetyt työkalut ja menetelmät.

Itse työn toteutus sujui suunnitellussa aikataulussa ja lopputulos oli mielestäni odotetun kaltainen, vaikka joitain kompromisseja työn aikana jouduttiinkin tekemään. Molemmille alustoille sain musiikkisoittimet toteutettua, molemmat sisältävät pääasiassa samat toiminnallisuudet eikä mitään dramaattisia ongelmia jäänyt kumpaankaan.

Jatkokehitysmielessä voisi miettiä Maemo-soittimen muuttamista toimimaan tulevassa Nokia N9 -MeeGo-laitteessa ja julkaista sovelluksen tulevien käyttäjien ladattavaksi. Myös WP7-soittimen kehitystä voisi jatkaa tutkimalla lisää alustan tarjoamia mahdollisuuksia kuvien hallintaan ja sitä kautta kehittää vähän sovelluksen visuaalista ilmettä. Lopuksi voitaisiin jopa julkaista sovellus käyttäjien ladattavaksi.

## LÄHTEET

App hub registration walkthrough. 2011. Microsoft. Saatavissa:  
[http://create.msdn.com/en-US/home/about/developer\\_registration\\_walkthrough](http://create.msdn.com/en-US/home/about/developer_registration_walkthrough).  
Hakupäivä 17.8.2011.

Application Platform Overview for Windows Phone. 2011. Microsoft.  
Saatavissa: [http://msdn.microsoft.com/en-us/library/ff402531\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/ff402531(v=VS.92).aspx).  
Hakupäivä 29.7.2011.

Get started with Maemo 5. 2009. Nokia Oyj. Saatavissa:  
[http://www.developer.nokia.com/document/Get\\_Started\\_with\\_Maemo\\_5/](http://www.developer.nokia.com/document/Get_Started_with_Maemo_5/).  
Hakupäivä 1.8.2011.

Ketterät käytännöt.fi 2008. Scrum. Saatavissa:  
<http://www.ketteratkaytannot.fi/Menetelmat/Scrum/>. Hakupäivä 1.8.2011.

Lee, Henry – Chuvyrov, Eugene 2010. Beginning Windows Phone 7  
Development. USA: Apress.

Lehtiniitty, Markus 2009. Nokia toi Qt:n virallisesti Maemo 5:lle. Saatavissa:  
[http://www.puhelinvertailu.com/uutiset.cfm/2009/10/09/nokia\\_toi\\_qt\\_n\\_virallises](http://www.puhelinvertailu.com/uutiset.cfm/2009/10/09/nokia_toi_qt_n_virallises)  
[ti\\_maemo\\_5\\_lle](http://www.puhelinvertailu.com/uutiset.cfm/2009/10/09/nokia_toi_qt_n_virallises). Hakupäivä 16.8.2011.

Lehto, Tero 2011. Nokia N9 teki vaikutuksen, mutta kysymyksiä jäi. Saatavissa:  
<http://blogit.tietokone.fi/tietojakoneesta/2011/06/nokia-n9-teki-vaikutuksen/>.  
Hakupäivä 17.8.2011.

Lindström, Jukka 2011. Scrum. Saatavissa: <http://reaktor.fi/osaaminen/scrum/>.  
Hakupäivä 1.8.2011.

Maemo background. 2011. Nokia Oyj. Saatavissa:  
<http://maemo.nokia.com/maemo/>. Hakupäivä 20.6.2011.



Maemo.org 2011. Saatavissa: <http://www.maemo.org>. Hakupäivä 20.6.2011.

Mountain Goat Software 2011. Saatavissa:

<http://www.mountangoatsoftware.com>. Hakupäivä: 17.8.2011.

Ojanperä, Veijo 2010. Nokia tappaa Maemon, tilalle Meego. Saatavissa:

[http://www.tietokone.fi/uutiset/nokian\\_maemo\\_ja\\_intelin\\_moblin\\_yhdistyvat](http://www.tietokone.fi/uutiset/nokian_maemo_ja_intelin_moblin_yhdistyvat).

Hakupäivä 17.8.2011.

Ovi Publisher Guide. 2011. Nokia Oyj. Saatavissa:

[https://p.d.ovi.com/p/g/ovistore\\_static/docs/Publisher\\_Guide.pdf?q=zVGvXkjWCt7piHnAaZYgVw](https://p.d.ovi.com/p/g/ovistore_static/docs/Publisher_Guide.pdf?q=zVGvXkjWCt7piHnAaZYgVw). Hakupäivä 16.8.2011.

Qt – Tools. 2011. Nokia Oyj. Saatavissa

<http://www.developer.nokia.com/Develop/Qt/Tools/>. Hakupäivä 17.8.2011.

Schwaber, Ken – Sutherland, Jeff 2011. Scrum. Suom. Lare Lekman sekä Arto Eskelinen, Petri Heiramo, Antti Järvinen, Lasse Koskela, Sirkka Lekman, Samuli Ruuskanen, Marko Taipale, Pentti Virtanen, Vesa Vänskä ja Lasse Ziegler. Saatavissa: <http://ketteryys.files.wordpress.com/2010/03/scrum-guide-fi.pdf>. Hakupäivä 18.8.2011.